



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 10320569

(43)Date of publication of application: 04.12.1998

(51)Int.Cl.

G06T 11/00
G06T 15/00

(21)Application number: 09126692

(71)Applicant:

SONY COMPUTER ENTERTAINMENT:KK

(22)Date of filing: 16.05.1997

(72)Inventor:

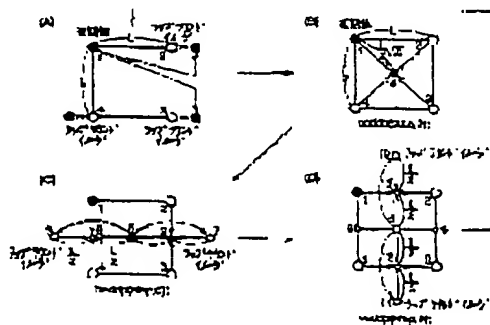
OBA AKIO

(54) IMAGE PROCESSING DEVICE, METHOD THEREFOR, RENDERING DEVICE AND METHOD THEREFOR

(57)Abstract:

PROBLEM TO BE SOLVED: To display an image high in quality by calculating other pixels which are distant from the 1st to 4th pixels by a prescribed distance based on these pixels and distances.

SOLUTION: A drawing processing part sets the same value as the initial value of a pixel 1 as pixels 2 to 4 of the positions, which are distant from a pixel 1 by prescribed distances in both horizontal and vertical directions. The value of a pixel 5 that is positioned at the center of pixels 1 to 4 is generated by adding a fluctuation component to the mean value of pixels 1 to 4. The fluctuation component is prescribed so as to be proportional to a prescribed random number and the distances between the pixels 1 to 4 and the pixel 5. The same value as the pixel 5 is set to the pixels 6 and 7 and the pixels 10 and 11 which are distant from the pixel 5 by a distance L in the horizontal or vertical direction. Then the value of pixels 8 and 9 and pixels 12 and 13 are calculated in the same way as the pixel 5. Such processing is repeated for generation of a magnified texture.



LEGAL STATUS

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-320569

(43) 公開日 平成10年(1998)12月4日

(51) IntCl⁵

G 0 6 T 11/00

15/00

識別記号

F I

G 0 6 F 15/72

3 5 0

4 5 0 A

審査請求 未請求 請求項の数11 O L (全 13 頁)

(21) 出願番号 特願平9-126692

(22) 出願日 平成9年(1997)5月16日

(71) 出願人 395015319

株式会社ソニー・コンピュータエンタテインメント

東京都港区赤坂7-1-1

(72) 発明者 大場 幸男

東京都港区赤坂7丁目1番1号 株式会社
ソニー・コンピュータエンタテインメント
内

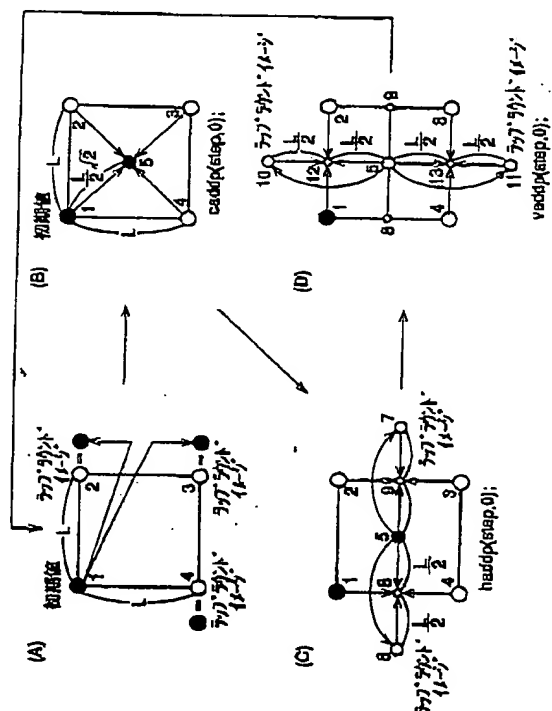
(74) 代理人 弁理士 稲本 義雄

(54) 【発明の名称】 画像処理装置および方法、並びにレンダリング装置および方法

(57) 【要約】

【課題】 高品位の画像を表示できるようにする。

【解決手段】 拡大テクスチャを生成するとき、初期値の画素を四隅に配置して、その中心の画素を演算により求める。このとき、演算される画素値を4つの画素の平均値と揺らぎ成分の合成として求める。揺らぎ成分は、各画素からの距離と乱数との積に比例させる。新たに生成された中心の画素を水平方向または垂直方向にそれぞれ配置して、そのとき中心に位置する画素を同様に生成する。



【特許請求の範囲】

【請求項1】 所定の基準となる第1の画素を設定する設定手段と、

前記第1の画素に対応する第2乃至第4の画素を、前記第1の画素から水平方向、垂直方向、または斜め方向に、所定の距離だけ離れた位置に配置する配置手段と、前記第1乃至第4の画素から所定の距離だけ離れている所定の位置の他の画素を、前記第1乃至第4の画素と、前記距離から演算する演算手段とを備えることを特徴とする画像処理装置。

【請求項2】 前記演算手段は、前記第1乃至第4の画素から、その中心に位置する第5の画素を演算することを特徴とする請求項1に記載の画像処理装置。

【請求項3】 前記演算手段は、前記第1乃至第4の画素の平均値に、乱数と前記距離により規定される値を加算して、前記第5の画素を演算することを特徴とする請求項2に記載の画像処理装置。

【請求項4】 前記演算手段は、前記第5の画素に対応する第6または第7の画素を、前記第5の画素から水平方向に、所定の距離だけ離れた位置に配置する水平配置手段と、

前記第1の画素、第4の画素、第5の画素、および第6の画素から、その中心に位置する第8の画素を演算するとともに、前記第2の画素、第3の画素、第5の画素、および第7の画素から、その中心に位置する第9の画素を演算する水平演算手段とをさらに備えることを特徴とする請求項3に記載の画像処理装置。

【請求項5】 前記演算手段は、前記第5の画素に対応する第10または第11の画素を、前記第5の画素から垂直方向に、所定の距離だけ離れた位置に配置する垂直配置手段と、

前記第1の画素、第2の画素、第5の画素、および第10の画素から、その中心に位置する第12の画素を演算するとともに、前記第3の画素、第4の画素、第5の画素、および第11の画素から、その中心に位置する第13の画素を演算する垂直演算手段とをさらに備えることを特徴とする請求項4に記載の画像処理装置。

【請求項6】 所定の基準となる第1の画素を設定する設定ステップと、

前記第1の画素に対応する第2乃至第4の画素を、前記第1の画素から水平方向、垂直方向、または斜め方向に、所定の距離だけ離れた位置に配置する配置ステップと、

前記第1乃至第4の画素から所定の距離だけ離れている所定の位置の他の画素を、前記第1乃至第4の画素と、前記距離から演算する演算ステップとを備えることを特徴とする画像処理方法。

【請求項7】 オリジナルのテクスチャを提供する第1の提供手段と、

前記オリジナルのテクスチャを拡大した拡大テクスチャ

を修整する修整テクスチャを提供する第2の提供手段と、

前記第1の提供手段の提供するオリジナルのテクスチャ、または前記第2の提供手段の提供する修整テクスチャを用いて、表示する画像のビットマップデータを描画する描画手段とを備えることを特徴とするレンダリング装置。

【請求項8】 前記修整テクスチャは、平均値が0となる符号付きカラーデータを有することを特徴とする請求項7に記載のレンダリング装置。

【請求項9】 前記修整テクスチャは、平均値が0となる加算用のカラーデータと減算用のカラーデータを有することを特徴とする請求項7に記載のレンダリング装置。

【請求項10】 前記修整テクスチャは、ブレンディング後のデスティネーションカラーの平均値が、ブレンディング前と同一となるように、その値が0と、1より大きい所定の値との間の所定の値をとる α パターンを有することを特徴とする請求項7に記載のレンダリング装置。

【請求項11】 オリジナルのテクスチャを提供する第1の提供ステップと、

前記オリジナルのテクスチャを拡大した拡大テクスチャを修整する修整テクスチャを提供する第2の提供ステップと、

前記第1の提供ステップの提供するオリジナルのテクスチャ、または前記第2の提供ステップの提供する修整テクスチャを用いて、表示する画像のビットマップデータを描画する描画ステップとを備えることを特徴とするレンダリング方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、画像処理装置および方法、並びにレンダリング装置および方法に関し、特に、より高品位の画像を表示することができるようにした画像処理装置および方法、並びにレンダリング装置および方法に関する。

【0002】

【従来の技術】図16は、コンピュータゲーム装置におけるレンダリングの例を表している。この例においては、奥行き方向に伸びる床102上に、オブジェクト100とオブジェクト101が表示されている。オブジェクト100は、近い位置に配置され、オブジェクト101は、遠い位置に配置されている。

【0003】床102を所定のテクスチャで表示するために、オリジナルテクスチャと、それを縮小したMIP-MAPテクスチャが予め用意されている。

【0004】そして、例えば床102の遠方のポリゴンP11は、例えばMIP-MAPテクスチャの中の、所定の縮小率のテクスチャT₁で表示される。また、床102の中

間の位置に位置するポリゴン P_{12} は、オリジナルテキストのテキスト T_7 で表示される。床102の最も手前のポリゴン P_{13} には、オリジナルテキストのうちの所定のテキスト T_8 が所定の拡大率で拡大して表示される。

【0005】このように、床102に所望のテキストを表示することで、リアルな画像とすることができる。

【0006】

【発明が解決しようとする課題】ところで、テキストマッピングされたポリゴンは、オリジナルテキストより拡大されると、画質が劣化する課題があった。例えば、拡大処理を画素を大きくすることで行うと（ポイントサンプリングすると）、画像がモザイク状となる。また、拡大処理をバイリニア補間で行うと、補間のため画像がぼける課題があった。

【0007】さらに、バイリニア補間の方がポイントサンプリングに較べて比較的良好の画像を得ることができるが、拡大処理は、より近景の画像ほど大きな拡大率で行われることになるため、近景ほどぼけた、コントラストの小さな画像となる。その結果、図16に示すように、近景に等倍のオブジェクト100が位置すると、このオブジェクト100は、テキストマッピングを利用していないため、鮮明な画像となっているのに対して、ほぼ同位置に位置する床102の画像が、ぼけた画像となってしまう、全体的に矛盾のある画像となり、見るものに違和感を与える結果となっていた。

【0008】本発明はこのような状況に鑑みてなされたものであり、より高品位の画像を表示できるようにするものである。

【0009】

【課題を解決するための手段】請求項1に記載の画像処理装置は、所定の基準となる第1の画素を設定する設定手段と、第1の画素に対応する第2乃至第4の画素を、第1の画素から水平方向、垂直方向、または斜め方向に、所定の距離だけ離れた位置に配置する配置手段と、第1乃至第4の画素から所定の距離だけ離れている所定の位置の他の画素を、第1乃至第4の画素と、距離から演算する演算手段とを備えることを特徴とする。

【0010】請求項6に記載の画像処理方法は、所定の基準となる第1の画素を設定する設定ステップと、第1の画素に対応する第2乃至第4の画素を、第1の画素から水平方向、垂直方向、または斜め方向に、所定の距離だけ離れた位置に配置する配置ステップと、第1乃至第4の画素から所定の距離だけ離れている所定の位置の他の画素を、第1乃至第4の画素と、距離から演算する演算ステップとを備えることを特徴とする。

【0011】請求項7に記載のレンダリング装置は、オリジナルのテキストを提供する第1の提供手段と、オリジナルのテキストを拡大した拡大テキストを修整する修整テキストを提供する第2の提供手段と、第1

の提供手段の提供するオリジナルのテキスト、または第2の提供手段の提供する修整テキストを用いて、表示する画像のビットマップデータを描画する描画手段とを備えることを特徴とする。

【0012】請求項11に記載のレンダリング方法は、オリジナルのテキストを提供する第1の提供ステップと、オリジナルのテキストを拡大した拡大テキストを修整する修整テキストを提供する第2の提供ステップと、第1の提供ステップの提供するオリジナルのテキスト、または第2の提供ステップの提供する修整テキストを用いて、表示する画像のビットマップデータを描画する描画ステップとを備えることを特徴とする。

【0013】請求項1に記載の画像処理装置および請求項6に記載の画像処理方法においては、第1の画素に対応する第2乃至第4の画素が、第1の画素から水平方向、垂直方向、または斜め方向に、所定の距離だけ離れた位置に配置される。そして、第1乃至第4の画素から所定の距離だけ離れている所定の位置の他の画素が、第1乃至第4の画素と、それらとの距離から演算される。

【0014】請求項7に記載のレンダリング装置および請求項11に記載のレンダリング方法においては、オリジナルのテキストまたは修整テキストを用いて、描画が行われる。

【0015】

【発明の実施の形態】以下に本発明の実施の形態を説明するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態（但し一例）を付加して本発明の特徴を記述すると、次のようになる。但し勿論この記載は、各手段を記載したものに限定することを意味するものではない。

【0016】請求項1に記載の画像処理装置は、所定の基準となる第1の画素を設定する設定手段（例えば図3のステップS1）と、第1の画素に対応する第2乃至第4の画素を、第1の画素から水平方向、垂直方向、または斜め方向に、所定の距離だけ離れた位置に配置する配置手段（例えば図3のステップS2）と、第1乃至第4の画素から所定の距離だけ離れている所定の位置の他の画素を、第1乃至第4の画素と、距離から演算する演算手段（例えば図3のステップS3）とを備えることを特徴とする。

【0017】請求項4に記載の画像処理装置は、演算手段は、第5の画素に対応する第6または第7の画素を、第5の画素から水平方向に、所定の距離だけ離れた位置に配置する水平配置手段（例えば図3のステップS4）と、第1の画素、第4の画素、第5の画素、および第6の画素から、その中心に位置する第8の画素を演算するとともに、第2の画素、第3の画素、第5の画素、および第7の画素から、その中心に位置する第9の画素を演算する水平演算手段（例えば図3のステップS5）とを

さらに備えることを特徴とする。

【0018】請求項5に記載の画像処理装置は、演算手段は、第5の画素に対応する第10または第11の画素を、第5の画素から垂直方向に、所定の距離だけ離れた位置に配置する垂直配置手段（例えば図3のステップS6）と、第1の画素、第2の画素、第5の画素、および第10の画素から、その中心に位置する第12の画素を演算するとともに、第3の画素、第4の画素、第5の画素、および第11の画素から、その中心に位置する第13の画素を演算する垂直演算手段（例えば図3のステップS7）とをさらに備えることを特徴とする。

【0019】請求項7に記載のレンダリング装置は、オリジナルのテクスチャを提供する第1の提供手段（例えば図12のテクスチャ領域72Bの領域72B₂）と、オリジナルのテクスチャを拡大した拡大テクスチャを修整する修整テクスチャを提供する第2の提供手段（例えば図12のテクスチャ領域72Bの領域72B₃）と、第1の提供手段の提供するオリジナルのテクスチャ、または第2の提供手段の提供する修整テクスチャを用いて、表示する画像のビットマップデータを描画する描画手段（例えば図2のグラフィックスプロセッサ71）とを備えることを特徴とする。

【0020】図1は、本発明の画像処理装置を応用したコンピュータゲーム装置の構成例を示すブロック図であり、このコンピュータゲーム装置は、3次元グラフィックス機能と動画再生機能とを備えている。

【0021】このコンピュータゲーム装置は、メインバス10と、サブバス20とからなる2つのシステムバスを備える構成を有している。これらメインバス10と、サブバス20との間のデータのやり取りは、バスコントローラ30により制御される。

【0022】そして、メインバス10には、メインCPU11、メインメモリ12、画像伸長デコード部13、前処理部14、描画処理部15、および、メインのDMAコントローラ16が接続されている。描画処理部15には、処理用メモリ17が接続されている。また、この描画処理部15は表示データ用のフレームバッファ（フレームメモリ）と、D/A変換回路を含み、この描画処理部15からのアナログビデオ信号がビデオ出力端子18に出力される。図示しないが、このビデオ出力端子18は、表示装置としての例えばCRTディスプレイに接続される。

【0023】サブバス20には、サブCPU21、サブメモリ22、ブートROM23、サブのDMAコントローラ24、音声処理用プロセッサ25、入力部26、補助記憶装置部27、および、拡張用の通信インタフェース部28が接続されている。補助記憶装置部27は、この例ではCD-ROMデコーダ41とCD-ROMドライバ42を備える。ブートROM23には、ゲーム機としての立ち上げを行うためのプログラムが格納されている。また、音声処理用

プロセッサ25に対しては、音声処理用メモリ25Mが接続されている。そして、この音声処理用プロセッサ25はD/A変換回路を備え、アナログ音声信号を音声出力端子29に出力する。

【0024】そして、補助記憶装置部27は、CD-ROMドライバ42に装填されたCD-ROMディスク6に記録されているアプリケーションプログラム（例えばゲームのプログラム）やデータをデコードする。CD-ROMディスク6には、例えば離散コサイン変換(DCT)を用いたMPEG2方式により画像圧縮された動画や静止画の画像データや、ポリゴンを修整するためのテクスチャ画像の画像データも記録されている。また、CD-ROMディスク6のアプリケーションプログラムには、ポリゴン描画命令が含まれている。

【0025】入力部26は、コントロールパッド26A、ビデオ信号の入力端子26B、および、音声信号の入力端子26Cを備えるものである。

【0026】メインCPU11は、メインバス10側の各部の管理および制御を行う。また、このメインCPU11は、物体を多数のポリゴンの集まりとして描画する場合の処理の一部を行う。メインCPU11は、1画面分の描画画像を生成するための描画命令列をメインメモリ12上に作成する。メインCPU11とメインバス10とのデータのやり取りは、データをバケット形式にしてバケット単位に行い、バースト転送を可能にしている。

【0027】メインメモリ12は、動画や静止画の画像データに対しては、圧縮された画像データのメモリ領域と、伸長デコード処理された伸長画像データのメモリ領域とを備えている。また、メインメモリ12は、描画命令列などのグラフィックスデータのメモリ領域（これをバケットバッファという）を備える。このバケットバッファは、メインCPU11による描画命令列の設定と、描画命令列の描画処理部15への転送とに使用される。

【0028】画像伸長デコード部13は、CD-ROMディスク6から再生され、メインメモリ12に転送された圧縮動画データやメインメモリ12上の圧縮されたテクスチャパターンデータの伸長処理を行うもので、この例では、MPEG2の画像圧縮方式を採用するので、それに対応したデコーダの構成を有している。

【0029】この画像伸長デコード部13の出力段には、瞬時の、つまりほぼリアルタイムでの可逆圧縮/伸長が可能で、圧縮率が例えば1/4乃至1/2程度の瞬時圧縮部50が設けられている。また、この例の画像伸長デコード部13は、その出力画像データの出力形式として、画像データの各画素の値を再量子化して出力する第1の出力データ形式（以下、この第1の出力データ形式をダイレクトカラー形式という）と、前記各画素の色が近似する色を示すインデックスデータに変換して出力する第2の出力データ形式（以下、この第2の出力

データ形式をインデックスカラー形式という)とを、描画処理部15での処理に適合させて選択可能としている。

【0030】描画処理部15は、メインメモリ12から転送されてくる描画命令を実行して、その結果をフレームメモリ(図2のVRAM72)に書き込む。フレームメモリから読み出された画像データは、D/A変換器を介してビデオ出力端子18に出力され、CRTなどの画像モニタ装置の画面に表示される。

【0031】なお、描画処理部15は、メインメモリ12から受け取る画像データの出力形式がインデックスカラー形式の場合にあっては、各画素データをインデックスデータから対応する代表色データに変換する処理を行う。このために、描画処理部15は、インデックスデータと代表色データとの変換テーブルであるCLUT(Color Look Up Table)を格納できるようにしている。

【0032】前処理部14は、CPUを備えるプロセッサの構成とされるもので、メインCPU11の処理の一部を分担することができるようにするものである。例えばポリゴンデータを、表示のための2次元座標データに変換する処理も、この前処理部14が行う場合がある。

【0033】そして、この例では、前処理部14とメインバス10との間に、瞬時圧縮部50による瞬時圧縮を解凍する瞬時解凍部60が設けられる。

【0034】次に、このゲーム装置の基本的な処理について以下に、簡単に説明する。

【0035】図1のコンピュータゲーム装置に電源が入入され、ゲーム装置本体1にCD-ROMディスク6が装填されると、ブートROM23の、ゲームを実行するためのいわゆる初期化処理をするためのプログラムが、サブCPU21により実行される。すると、CD-ROMディスク6の記録データが次のようにして取り込まれる。

【0036】すなわち、補助記憶装置部27においては、CD-ROMディスク6から、圧縮画像データ、描画命令およびメインCPU11が実行するプログラムが、CD-ROMドライバ42、CD-ROMデコーダ41を介して読み出され、サブDMAコントローラ24によってサブメモリ22に一旦ロードされる。

【0037】そして、このサブメモリ22に取り込まれたデータは、サブDMAコントローラ24およびバスコントローラ30、さらにはメインDMAコントローラ16によってメインメモリ12に転送される。なお、サブCPU21は、描画処理部15のフレームに対して直接的にアクセスできるように構成されており、このサブCPU21によっても表示画像内容の変更が、描画処理部15の制御とは離れて可能とされている。

【0038】図2は、描画処理部15の基本的な構成例を表している。この描画処理部15は、グラフィックプロセッサ71、VRAM72、およびCRTコントローラ73より構成されている。グラフィックプロセッサ71は、

図3のフローチャートを参照して後述するフラクタル生成アルゴリズムを実行するプログラムを内蔵している。VRAM72は、テクスチャ素材を記録する領域72A、生成されたテクスチャが描画される領域72B、図示せぬCRTに出力するビットマップデータが交互に描画される領域72C、72Dを有している。

【0039】グラフィックプロセッサ71は、メインメモリ12からの描画命令を実行して、領域72Cまたは領域72Dに、交互にビットマップデータを描画する。領域72Cにビットマップデータが新たに書き込まれているとき、領域72Dに既に書き込まれているビットマップデータが読み出され、CRTコントローラ73からCRTに出力され、表示される。領域72Cへの描画が完了すると、今度は領域72Cからビットマップデータが読み出され、表示される。そして、このとき、領域72Dに新たな画像が描画される。このように、領域72Cと領域72Dは、書き込みと読み出しの動作が交互に行われる。

【0040】テクスチャ領域72Bには、図16に示した場合と同様に、オリジナルテクスチャとMIP-MAPテクスチャが記憶されており、適宜所定のものが読み出され、領域72Cまたは72Dに描画される。グラフィックプロセッサ71は、テクスチャを拡大する必要がある場合、内蔵するフラクタル生成アルゴリズムを用いて、フラクタルを生成し、テクスチャ素材領域72Aの所定の領域に記憶させる。

【0041】次に、このアルゴリズムを図3を参照して説明する。最初にステップS1において、グラフィックプロセッサ71は、所定の画素データを初期値として設定する。また、このとき、フラクタル生成するテクスチャの大きさに対応する距離Lが初期設定される。

【0042】次に、ステップS2に進み、グラフィックプロセッサ71は、所定の位置にラップラウンドイメージを設定する処理を行う。すなわち、図4(A)に示すように、ステップS1で設定した初期値の画素を画素1とすると、水平方向に距離Lだけ離れた位置の画素2、垂直方向に距離Lだけ離れた位置の画素4、および画素2から距離Lだけ垂直方向に離れた位置(画素4から水平方向に距離Lだけ離れた位置)の画素3として、画素1の初期値と同一の値を設定する。

【0043】次に、ステップS3に進み、ステップS2で設定した4つの画素(すなわち、画素1乃至画素4)から、その中心の画素の値を演算する。

【0044】すなわち、図4(B)に示すように、画素1乃至画素4の中心に位置する画素5の値を演算する。この画素5の値hは、次式で表される。

$$h = (dt1 + dt2 + dt3 + dt4) / 4 + q \times R \times TD \times hstep$$

【0045】なお、上記式において、dt1乃至dt4は、画素1乃至画素4の値を意味し、qは、発生された

乱数を表し、RTDは、 $2^{1/2}$ を表し、hstepは、距離(いまの場合、 $L/2$)を表している。

【0046】すなわち、画素5の値hは、画素1乃至画素4の平均値に揺らぎ成分を加算して生成されている。この揺らぎ成分は、-1.0から1.0の間の乱数qと、画素1乃至画素4から画素5までの距離RTD×hstep($2^{1/2} \cdot L/2$)に比例するように規定されている。これにより、自己相似性が実現され、フラクタル演算ができるようになされている。

【0047】次に、ステップS4に進み、グラフィックプロセッサ71は、水平方向に対して、ラップラウンドイメージを設定する処理を実行する。すなわち、図4

(C)に示すように、画素5から水平方向に距離Lだけ離れた位置の画素6、または画素7として、画素5と同一の値を設定する。

【0048】次に、ステップS5に進み、新たな4点の中心に配置されている画素データをフラクタル演算する処理を実行する。すなわち、画素1、画素4、画素5、および画素6の中心に位置する画素8のデータが演算されるとともに、画素2、画素3、画素5、および画素7の中心に位置する画素9の画素値が求められる。

【0049】この場合、例えば画素8の画素1、画素4、画素5、および画素6からの距離は、 $L/2$ であるため、次式から演算される。

$$h = (dt1 + dt4 + dt5 + dt6) / 4 + q \times hstep$$

【0050】次に、ステップS6に進み、グラフィックプロセッサ71は、垂直方向に対して、ラップラウンドイメージを設定する。すなわち、図4(D)に示すように、画素5から垂直方向に距離Lだけ離れた位置の画素10または画素11に、画素5と同一の値の画素を設定する。

【0051】次に、ステップS7に進み、新たに生成された4点の中心の画素データをフラクタル演算する処理が実行される。

【0052】すなわち、図4(D)に示すように、画素1、画素2、画素5、および画素10の中心に位置する画素12の値と、画素3、画素4、画素5、および画素11の中心に位置する画素13の値が演算される。

【0053】この場合における例えば画素12の値は、次式で演算される。

$$h = (dt1 + dt2 + dt5 + dt10) / 4 + q \times hstep$$

【0054】次に、ステップS8に進み、全ての画素データを生成したか否かを判定し、まだ生成していない画素データが残っている場合には、ステップS2に戻り、同様の処理が繰り返し実行される。

【0055】例えば、図4(D)に示すように、画素1乃至画素5、画素8、画素9、画素12、および画素13を生成した後、さらにより近接した位置の画素を求め

る場合には、例えば画素1、画素12、画素5、および画素8の4つの画素を選択し、その中心の画素を求め、さらに、水平方向、または垂直方向に中心の画素をラップラウンドして、新たな画素データを演算する。

【0056】以上のような処理を必要な回数繰り返すことで、所定の大きさの拡大テクスチャを生成することができる。

【0057】図5乃至図6は、以上のフラクタル生成アルゴリズムのより具体的なプログラムを表している。

10 【0058】以上のようにして生成した単位のテクスチャを複数個配列して、より大きなテクスチャを構成したとしても、各単位のテクスチャとの境界部が視認されてしまうようなことがない。すなわち、図7に示すように、生成した1つの単位としてのテクスチャの4つの角の画素データは、いずれも同一(図7の表示例の場合、A)となっている。従って、これを上下左右方向に配列したとしても、各単位の境界部分に配置される画素データは、実質的に同一のデータであるから、その境界部が視認されるようなことがない。

20 【0059】これに対して、図8に示すように、四隅に配置する画素を初期値と異なる値にすると、そのようにして生成された単位のテクスチャを複数個配列すると、各単位の境界部に配置される画素データがそれぞれ異なるものとなり、境界部が目立つものとなる。本願においては、上述したように、フラクタルパターンをサイクリックに生成するようにしたので、これが防止される。

【0060】図9は、以上のようにして生成したサイクリックテクスチャをさらに合成して、新たなサイクリックテクスチャを生成する様子を示している。同図に示すように、テクスチャT₁乃至T₃が、単位のサイクリックテクスチャを複数個配列することで生成されている。テクスチャT₄は、テクスチャT₁とT₃を合成して生成されている。テクスチャT₅は、テクスチャT₂とT₃を合成して生成されており、また、テクスチャT₆は、テクスチャT₁乃至T₃を合成して生成されている。

【0061】なお、このような2以上のテクスチャを合成して1つのテクスチャを生成する場合、各テクスチャの位相をずらすことで、微妙に異なる多くの種類のテクスチャを生成することができる。

40 【0062】なお、位相をずらす処理は、読み出しまたは書き込みアドレスをずらすことで実現することができる。

【0063】図10の表示例においては、テクスチャT₁₁が、テクスチャT₁とテクスチャT₃をそれぞれ異なる拡大率(縮小率)で合成して生成されている。同様に、テクスチャT₁₂は、T₂とT₃を異なる拡大率(縮小率)で合成して生成され、テクスチャT₁₃は、テクスチャT₁乃至T₃を、それぞれ異なる拡大率(縮小率)で合成することで、生成されている。このように、簡単なサイクリックパターンで、拡大率(縮小率)の最小公倍数の大

50

きさのパターンを合成することができる。

【0064】さらに、図11に示すように、サイクリックテクスチャ T_{21} と α 値により制御された抜き型の画像 T_{22} とを合成して、合成テクスチャ T_{23} を生成すると、より複雑なテクスチャパターンを合成することができる。

【0065】なお、 $\alpha=0$ の画素は、実質的に透明な画素を表し、 $\alpha=1$ の画素は、透明でない画素を表している。

【0066】以上のようにして、サイクリックテクスチャを生成すると、画像のモザイク模様あるいはボケを防止することができる。

【0067】次に、拡大テクスチャをより細密化する方法について、図12を参照して説明する。

【0068】すなわち、この場合、図12に示すように、VRAM72のテクスチャ領域72Bの領域72B₂にオリジナルテクスチャを記憶させるとともに、領域72B₁にMIP-MAPテクスチャを記憶させる。そして、上述した場合と同様にして、例えば床102の遠景のポリゴンP₁は、対応する縮小率のテクスチャ T_A を用いて表現する。また、中景のポリゴンP₂は、オリジナルテクスチャ T_B を用いて表現する。近景のポリゴンP₃は、例えばオリジナルテクスチャ T_C を拡大して表現する。以上の処理は、従来における場合と同様であるが、拡大テクスチャをより細密化するために、さらに、拡大テクスチャについては次のような修整処理を施す。

【0069】すなわち、テクスチャ領域72Bの領域72B₃には、拡大用修整テクスチャが記憶される。拡大テクスチャには、その拡大率に応じて、その拡大用修整テクスチャがさらにブレンディングされる。例えば、ポリゴンP₃には、対応する拡大率の拡大用修整テクスチャ T_D がブレンディングされる。

【0070】拡大用修整テクスチャとしては、例えば図13に示すように、符号付きカラーデータで、平均値が0となるように変化するテクスチャが記憶されている。その結果、例えばポリゴンP₃のテクスチャは、この拡大用修整テクスチャで修整（モジュレーション）されるため、より変化に富んだテクスチャとなる。但し、その変化の平均は0となっているため、隣接するポリゴンとの調和が図られ、1つのポリゴンだけが他のポリゴンから浮き出て見えるようなことが防止されている。

【0071】このような効果を出すには、例えば図14に示すように、拡大用修整テクスチャを、加算用のカラーデータと減算用のカラーデータとで構成し、加算用のカラーデータと減算用のカラーデータの和が0となるようにしておいてもよい。

【0072】あるいはまた、図15に示すように、通常、0と1の間の値である α ブレンディングにおける α の値を、1以上の値をも取り入れるものとし、この α の値で拡大用修整テクスチャを規定するようにしてもよ

い。この場合においても、デスティネーションカラーの平均値がブレンディング前と同一になるように α の値が設定される。すなわち、デスティネーションカラーの値をDsColorとすると、次式に示すような内挿処理が行われる。

$$DsColor \times \alpha + ScColor \times (1 - \alpha)$$

【0073】なお、ScColorは、ソースカラーを表す。

【0074】

10 【発明の効果】以上の如く、請求項1に記載の画像処理装置および請求項6に記載の画像処理方法によれば、第2乃至第4の画素を、第1の画素に対応して設定し、第1乃至第4の画素から所定の距離だけ離れている所定の位置の他の画素を、第1乃至第4の画素と、その距離から演算するようにしたので、生成した画像を配列した場合においても、その境目が見えない大きな画像をさらに生成することが可能となる。

【0075】請求項7に記載のレンダリング装置および請求項11に記載のレンダリング方法によれば、拡大テクスチャを修整する修整テクスチャを用いて、ビットマップデータを描画するようにしたので、拡大テクスチャが他のオブジェクトとの画像とともに表示されたような場合において、違和感のある画像となるのを防止することができる。

【図面の簡単な説明】

【図1】本発明の画像処理装置を応用したコンピュータゲーム装置の構成例を示すブロック図である。

【図2】図1の描画処理部15の構成例を示すブロック図である。

30 【図3】図2の構成例の動作を説明するフローチャートである。

【図4】図3の処理を説明する図である。

【図5】フラクタル生成アルゴリズムを適用したプログラムの具体的例を示す図である。

【図6】フラクタル生成アルゴリズムを適用したプログラムの具体的例を示す図である。

【図7】テクスチャを複数個配列した状態を説明する図である。

40 【図8】テクスチャを複数個配列した状態を説明する図である。

【図9】テクスチャの合成を説明する図である。

【図10】テクスチャの合成を説明する図である。

【図11】テクスチャの合成を説明する図である。

【図12】拡大用修整テクスチャを説明する図である。

【図13】拡大用修整テクスチャを表現する例を示す図である。

【図14】拡大用修整テクスチャを表現する例を示す図である。

50 【図15】拡大用修整テクスチャを表現する例を示す図である。

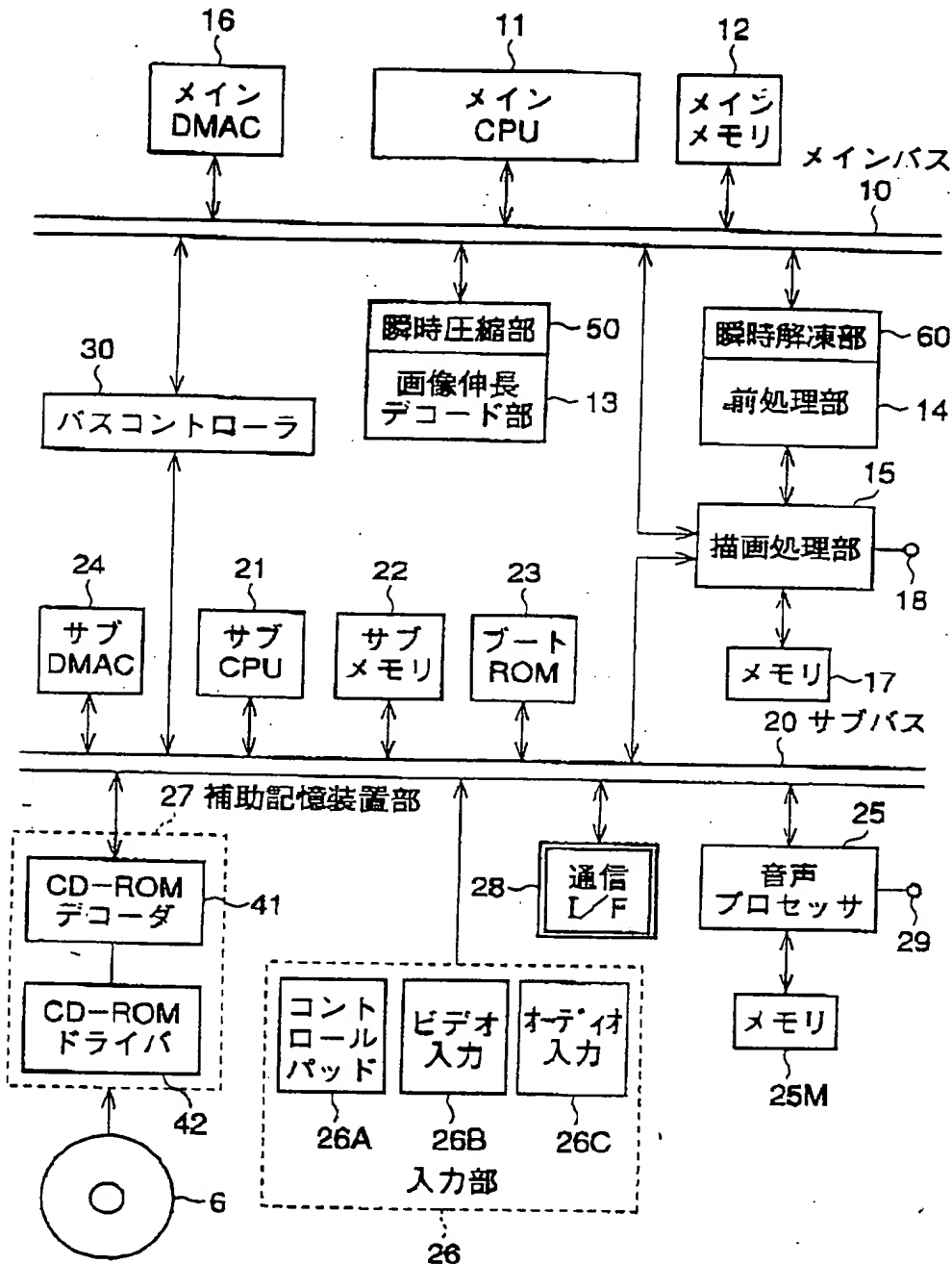
【図16】従来のテクスチャマッピングを説明する図である。

【符号の説明】

11 メインCPU, 12 メインメモリ, 14 前

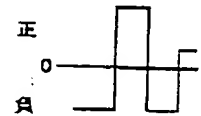
処理部, 15 描画処理部, 71 グラフィックプロセッサ, 72 VRAM, 73 CRTコントローラ, 72A乃至72D 領域

【図1】

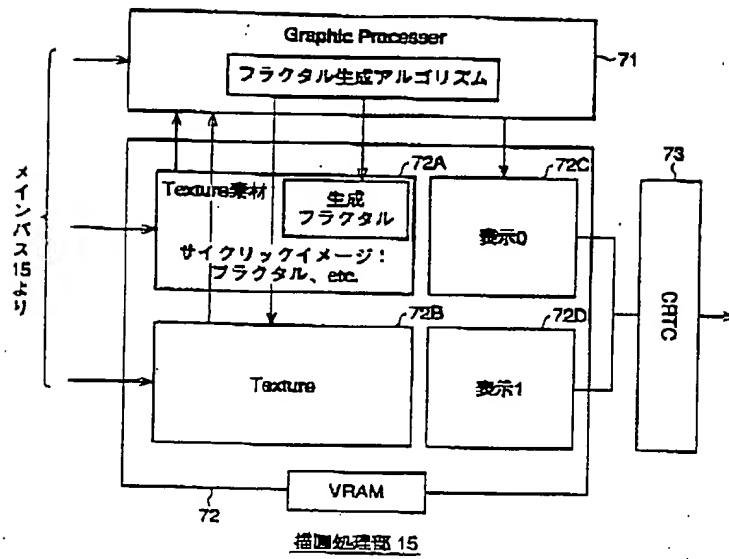


【図13】

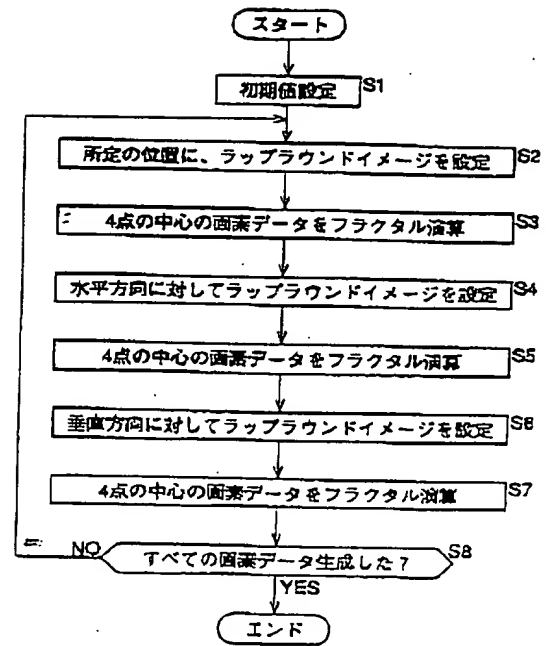
符号付きカラーデータ



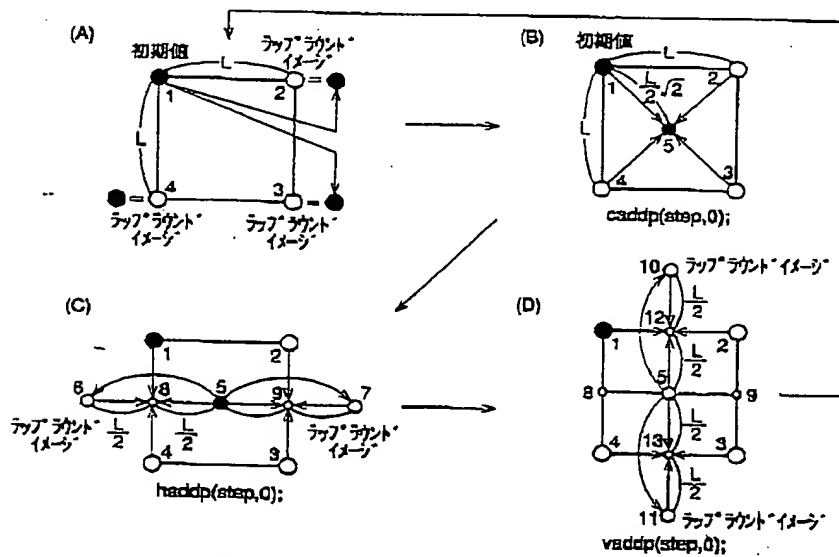
【図2】



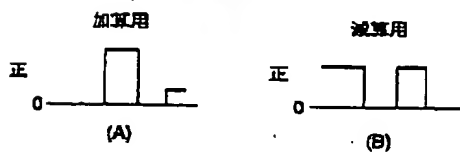
【図3】



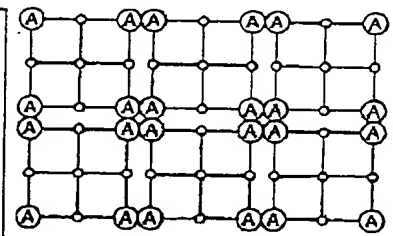
【図4】



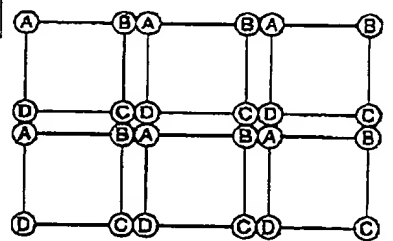
【図14】



【図7】



【図8】



【図5】

プログラム1

```

/*
 *
 *
 *
 *
 */
Fractal Generator (Akio Ohba)

Copyright (C) 1997 by Sony Computer Entertainment
All rights Reserved

#define N 7
#define LM (1<<N) 4096 /* LM=2^N */
#define ONE 5793 /* フラクタル配列の幅 */
#define RTD /* 固定小数点での1.0 */
#define FP 32 2048 /* 固定小数点でのSQRT(2.0) */
#define INIRAND 1800 /* フラクタル調整パラメータ */
#define INIFR /* 乱数初期値 */
/* フラクタル配列初期値 */

long MDATA; /* M系列発生関数のレジスタメモリ */
int geo_h[LM][LM]; /* フラクタル配列; Array size=LM * LM */

fractal()
{
    int i, step, x, y;

    smrand(INIRAND); /* Fractal Data Generator */
    geo_h[0][0]=INIFR; /* 乱数初期値 */
    /* フラクタル配列初期値 */

    for(i=0; i<N; i++){
        step=LM>>i; /* 斜め方向のデータ生成 */
        caddp(step,i); /* 横方向のデータ生成 */
        haddp(step,i); /* 縦方向のデータ生成 */
        vaddp(step,i);
    }

    unsigned long mrand() /* 31次のM系列発生関数 */
    {
        MDATA~=(MDATA|((MDATA&8)<<28)^((MDATA&1)<<31))>>1;
        return(MDATA);
    }

    smrand(i) /* M系列発生関数の初期値設定 */
    long i;
    {
        MDATA=i;
    }

    vaddp(step,i)
    int step,i;
    {
        int x,y,hstep,dt1,dt2,dt3,dt4,lx,rx,uy,dy;
        hstep=step>>1;
        for(x=0; x<LM; x+=step){
            for(y=hstep; y<LM; y+=step){
                lx=(x+LM-hstep)%LM;
                rx=(x+LM+hstep)%LM;
                uy=(y+LM-hstep)%LM;
                dy=(y+LM+hstep)%LM;
                dt1=geo_h[lx][uy]; /* 上方のデータ */
                dt2=geo_h[lx][dy]; /* 下方のデータ */
                dt3=geo_h[lx][y]; /* 左方のデータ */
                dt4=geo_h[rx][y]; /* 右方のデータ */
                geo_h[lx][y]=gendatahv(dt1,dt2,dt3,dt4,hstep,i);
            }
        }
    }
}

```

【図6】

```

haddp(step,i)
int step,i;
{
    int x,y,hstep,dt1,dt2,dt3,dt4,lx,rx,uy,dy;
    hstep=step >> 1;
    for (x=hstep; x<LM; x+=step) {
        for (y=0; y<LM; y+=step) {
            bx=(x+LM-hstep) % LM;
            rx=(x+LM+hstep) % LM;
            uy=(y+LM-hstep) % LM;
            dy=(y+LM+hstep) % LM;
            dt1=geo_h[bx][uy]; /* 上方のデータ */
            dt2=geo_h[bx][dy]; /* 下方のデータ */
            dt3=geo_h[lx][y]; /* 左方のデータ */
            dt4=geo_h[lx][y]; /* 上方のデータ */
            geo_h[x][y]=gendatahv(dt1,dt2,dt3,dt4,hstep,i);
        }
    }
}

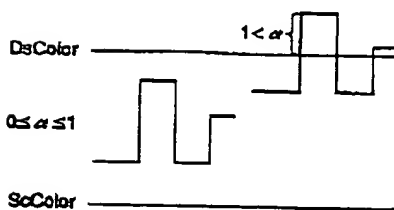
caddp(step,i)
int step,i;
{
    int x,y,hstep,dt1,dt2,dt3,dt4,lx,rx,uy,dy;
    hstep=step >> 1;
    for (x=hstep; x<LM; x+=step) {
        for (y=hstep; y<LM; y+=step) {
            bx=(x+LM-hstep) % LM;
            rx=(x+LM+hstep) % LM;
            uy=(y+LM-hstep) % LM;
            dy=(y+LM+hstep) % LM;
            dt1=geo_h[bx][uy]; /* 左上のデータ */
            dt2=geo_h[bx][dy]; /* 左下のデータ */
            dt3=geo_h[lx][uy]; /* 右上のデータ */
            dt4=geo_h[lx][dy]; /* 右下のデータ */
            geo_h[x][y]=gendatac(dt1,dt2,dt3,dt4,hstep,i);
        }
    }
}

int gendatahv(dt1,dt2,dt3,dt4,hstep,i)
int dt1,dt2,dt3,dt4,hstep,i;
{
    int h;
    int q;
    q=mrnd() & 0x1FFF; /* 13bit; (0<=q<8192)のM系列による乱数 */
    q=ONE-q; /* -1.0<q<1.0 */
    h=(dt1+dt2+dt3+dt4)/4+((q * hstep * FP) >> 12);
    return(h);
}

int gendatac(dt1,dt2,dt3,dt4,hstep,i)
int dt1,dt2,dt3,dt4,hstep,i;
{
    int h;
    int q;
    q=mrnd() & 0x1FFF; /* 13bit; (0<=q<8192)のM系列による乱数 */
    q=ONE-q; /* -1.0<q<1.0 */
    h=(dt1+dt2+dt3+dt4)/4+((RTD * hstep * FP) >> 12) >> 12; /* RTDによる距離補正 */
    return(h);
}

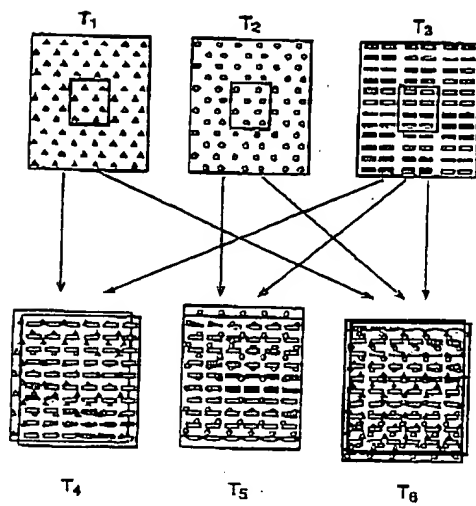
```

【図15】



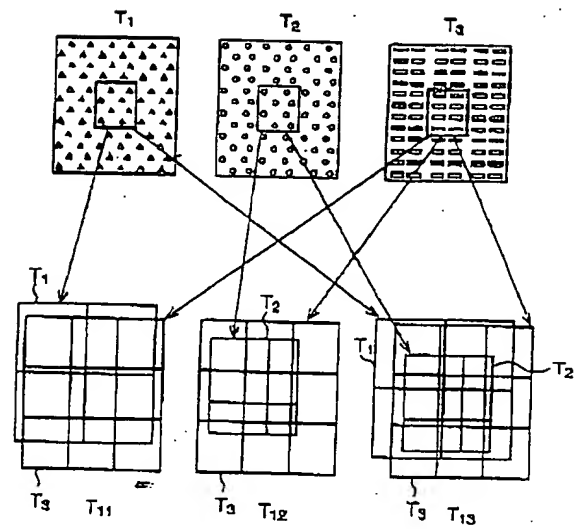
【図9】

サイクリックテクスチャ



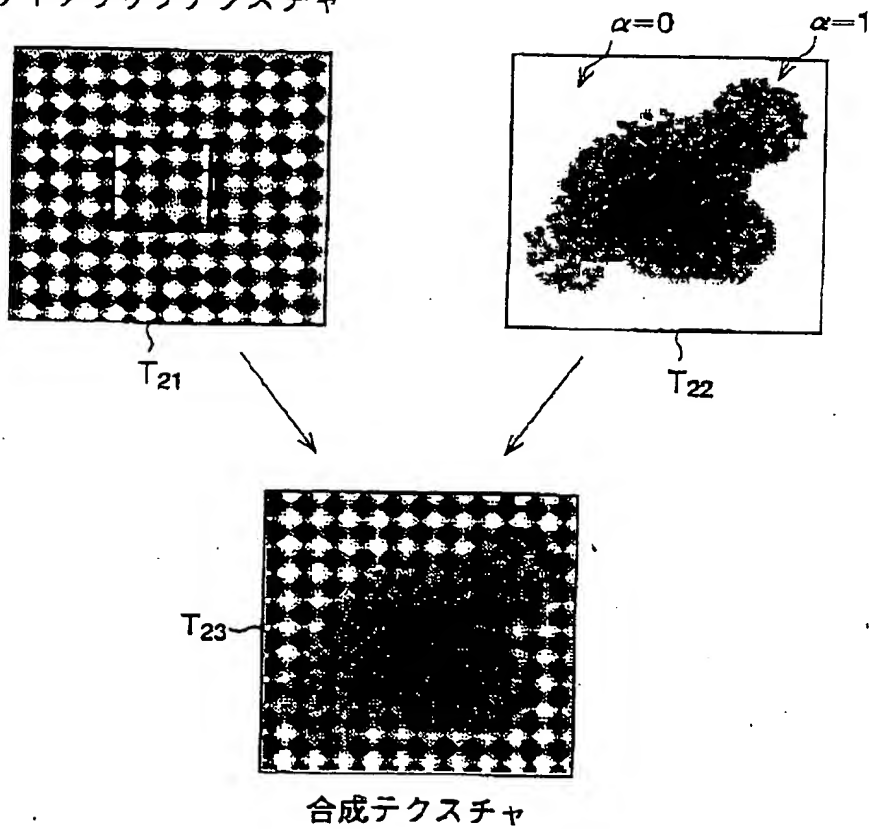
【図10】

サイクリックテクスチャ



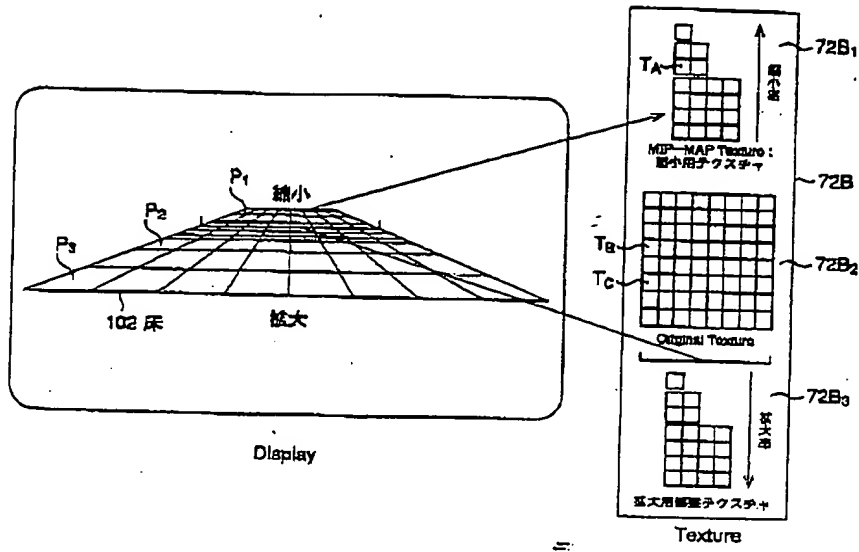
【図11】

サイクリックテクスチャ



合成テクスチャ

【図12】



【図16】

